

Td de langage C

Petits exercices

Travail de saisie, d'affichage avec des caractères, des entiers et petits algorithmes sur des entiers

1/ Écrire un programme qui calcule la somme de deux nombres entiers.

2/ Écrire un programme qui demande la saisie de deux nombres entiers et affiche leur somme.

3/ Idem 2/ avec des flottants

4/ Écrire un programme qui affiche les codes ASCII des lettres et des chiffres de 'A' à 'Z', 'a' à 'z', '1' à '9' sous la forme

caractère = A code = 65 code hexa = 41

caractère = 9 code = 57 code hexa = 39

5/ Écrire un programme qui demande la saisie de nombres entiers en affichant au fur et à mesure le plus grand et le plus petit. La saisie s'arrête quand le nombre 0 est entré. Les nombres saisis n'ont pas besoin d'être conservés en mémoire.

6/ Écrire un programme qui demande la saisie de deux nombres et divise le plus grand par le plus petit. Afficher le quotient et le reste. Vous ne disposez pas de l'opérateur / mais seulement de l'addition et la soustraction.

7/ Demander la saisie d'un entier et déterminer si cet entier est pair ou impair. Vous ne disposez que de l'opérateur de division entière /

8/ Demander la saisie d'un entier et déterminer s'il s'agit d'un nombre premier

a/ méthode séquentielle

b/ méthode récursive par dichotomie pour les grands nombres

9/ Demander la saisie d'un nombre entier et afficher tous les nombres premiers inférieurs à ce nombre.

Travail sur les chaînes de caractères

10/ Écrire un programme qui saisit une chaîne de caractère sur l'entrée standard, puis saisit un mot et regarde si le mot est dans la chaîne.

11/ Idem en demandant la saisie successive d'un nouveau mot. Le programme s'arrête quand le mot « exit » est entré.

Travail avec les tableaux

12/ Saisir 10 réels, les ranger dans un tableau. Calculer et afficher leur moyenne et leur écart-type.

Travail avec les fichiers

13/ Écrire un programme de recopie de fichier binaire, caractère par caractère.

Programmes de tri

14/ Demander la saisie au clavier d'un certain nombre d'entiers (ce nombre pourra lui-même être préalablement demandé. Ranger les entiers dans un tableau. Afficher le tableau. Classer ce tableau du plus petit entier au plus grand. Réafficher le tableau.

15/ Idem 14 mais avec des chaînes de caractères à la place des entiers. On définira une taille maximale pour les chaînes et on réservera dynamiquement la mémoire pour stocker chaque chaîne. Le tableau de chaînes sera de type `char **theTab` (pointeur sur des pointeurs de caractères).

16/ Idem 14 mais le programme de tri recevra la fonction de comparaison en argument. On pourra donc appliquer le même programme de tri avec différentes règles de classement.

Exercices plus conséquents

17/ Écrire un programme qui réalise une petite calculatrice

a/ sans mémoire

b/ avec mémoire

18/ Écrire un programme qui met en œuvre une fonction `grep` simple :

`grep chaîne chemin`

il s'agit de rechercher les occurrences de la chaîne de caractère 'chaîne' dans les fichiers du répertoire 'chemin' et dans toute la hiérarchie de ses sous-répertoires.

Dans un deuxième temps, chaîne ne sera plus une chaîne littérale à chercher mais un motif pouvant correspondre à plusieurs chaînes (voir : expressions régulières).

19/ Écrire un programme mettant en œuvre une structure de donnée qui se rapproche des « dictionnaires » du langage Python.

Exercices sur les matrices

20/ Proposez une structure de données pour décrire des matrices carrées de tailles quelconque (càd que le nombre de lignes et de colonnes ne sont pas fixés à l'avance mais sont donnés comme arguments de la fonction de création de la matrice).

21/ Écrire un programme qui saisie une matrice carrée, l'affiche à l'écran et libère l'espace dynamique qui a été utilisé pour stocker la matrice. Le programme demande d'abord la saisie des dimensions de la matrice (nombre de ligne, nombre de colonnes) puis il demande un à un chacun des coefficients successifs de la matrice.

22/ Écrire une fonction de multiplication de deux matrices carrées.

Remarque : il faut que le nombre de colonnes de la première soit égal au nombre de lignes de la seconde. La fonction de multiplication prendra en argument un pointeur vers une matrice vide. A l'intérieur de la fonction il faudra, si la multiplication est possible, réserver dynamiquement l'espace mémoire nécessaire au stockage de la matrice résultat.

Ce sera au programme appelant (le programme qui utilisera la fonction de multiplication des matrices) de libérer l'espace mémoire de stockage de la matrice résultat quand il n'en aura plus besoin.

23/ Idem exercice précédent mais cette fois on ne saisit plus les matrices à partir du clavier. Les matrices seront décrites dans un fichier texte externe .ini et lues dans ce fichier par le programme.